Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 5014355, 14 pages
https://doi.org/10.1155/2023/5014355

Research Article

ContextAD: Context-Aware Acronym Disambiguation with Siamese BERT Network

Lizhen Ou, Yiping Yao, Xueshan Luo, Xinmeng Li, and Kai Chen

College of System Engineering, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to Kai Chen; chenkai@nudt.edu.cn

Received 11 March 2023; Revised 20 June 2023; Accepted 20 July 2023; Published 29 July 2023

Academic Editor: B. B. Gupta

Acronym disambiguation is the process of determining the correct expansion of an acronym in given context, which can assist many downstream natural language processing tasks. Typically, existing methods on this task will directly perform semantic comparisons between the candidate expansions and the original sentence, ignoring the relevance of contextual information to expansions. To solve this issue, this paper proposes a context-aware acronym disambiguation method with Siamese BERT network (ContextAD). First, we combine each candidate expansion with corresponding acronym's context to form a new sentence set. Then, the new and original sentences are input into a Siamese BERT network that can obtain the semantic similarity. The new sentences and the separate candidate expansions are input into the Siamese BERT network, respectively, along with the original sentences, which can obtain another semantic similarity. Finally, the two different semantic similarities are combined to determine the most suitable expansion. We quantify the improvement of our proposed ContextAD model against a state-of-the-art baseline using the public dataset of the shared tasks of acronym disambiguation (AD) held under AAAI-2021 workshop on SDU and show that it achieves a better performance based on the same BERT model.

1. Introduction

Acronyms are shortened forms of longer phrases and are often used in writing, especially academic writing, to save space and streamline expression. However, in natural language processing tasks such as question answering, machine reading comprehension, information extraction [1], sensitive word detection, and retrieval, it is often necessary to use the definition of acronyms. Acronym disambiguation can provide an effective acronym comprehension scheme. Its purpose is to select the most appropriate definition from the acronym dictionary according to the meaning of the sentence containing the acronym. The sources of the acronym dictionary mainly include WEB data acquisition and manual construction.

Early studies mainly used the construction of acronym dictionaries using WEB information. For example, some researches directly obtain web pages containing acronym and definitions [2] or automatically extract acronyms and corresponding definitions from the interaction between users and network data [3]. Then, machine learning methods [1], pattern matching [4–6], and semantic network generation [7] are used to achieve acronym disambiguation. However, the level of network data is uneven, and it is not easy to ensure the quality of the acronym dictionary. Moreover, methods based on network data often require the device to be connected to the network and cannot be applied offline. Therefore, some scholars use artificially constructed dictionaries and machine learning algorithms for disambiguation [8]. Examples thereof are shown in Figure 1. The model needs to pick out accurate definitions based on acronyms and their contextual information from the corresponding dictionary.

After the development of recent years, dictionary-based acronym disambiguation methods have made great progress. Early researchers used statistical methods for feature extraction, such as support vector machines, naive Bayes, and k-nearest neighbors [9, 10]. This kind of method is simple, but it has low precision and recall. After machine learning algorithms, especially deep neural networks

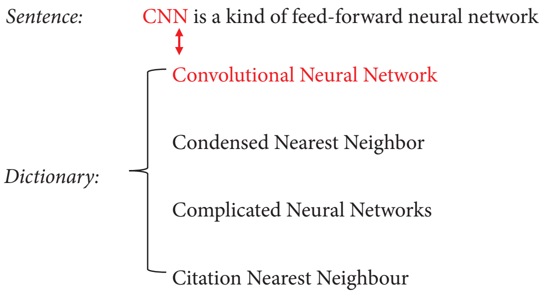

Sentence:
CNN is a kind of feed-forward neural network
Dictionary:
Convolutional Neural Network
Condensed Nearest Neighbor
Complicated Neural Networks
Citation Nearest Neighbour

Figure 1: A toy example of acronym disambiguation.

demonstrated powerful feature extraction, and neural network-based acronym recognition methods began to proliferate, e.g., convolution neural networks (CNNs) or long short-term memory (LSTM) [11]. However, traditional deep neural networks have difficulty in incorporating prior knowledge and can only extract features from the dataset. While the transformer-based methods represented by BERT and its derivative models are able to obtain features from a large amount of unlabelled data and apply these features (prior knowledge) to downstream tasks. However, traditional deep neural networks have difficulty in incorporating prior knowledge and can only extract features from the dataset, while the transformer-based methods represented by BERT [12] and its derivative models are able to obtain features from a large amount of unlabelled data and apply these features (prior knowledge) to downstream tasks. This approach can greatly improve the accuracy of the model. Singh and Kumar used the SpanBERT [13] model to transform the acronym disambiguation problem into a span prediction problem [14]. Pan et al. tried different BERT models and finally found that the SciBERT model [15] has a significant advantage in the acronym disambiguation task [16]. Weng et al. then used the DEBERTA [17, 18] model for their experiments [19], while Song et al. verified the validity of the T5 model [20] which is an alias of text-to-text transfer transformer, and the basic idea of this model is that all NLP problems can be defined as "text-to-text" problems, i.e., "input text and output text" [21].

Existing methods tend to analyse paraphrases directly with the original sentences, without taking full advantage of the similarity feature between paraphrases and acronyms. In this paper, we design a framework based on twin networks mainly based on the property that acronyms are completely alternative to exact paraphrases. Complete substitutability means that an accurate paraphrase can replace the acronym in the original text without changing the sentence paraphrase. This property is similar to the way humans think, and when they disambiguate acronyms, they usually choose to replace the acronym with the candidate paraphrase and analyse it in context to determine its suitability.

Therefore, this paper proposes a context-aware acronym disambiguation method with Siamese BERT network (ContextAD), which combines candidate paraphrases with the context of acronyms to form new sentences and uses the Siamese network model to obtain the similarity between the new sentence and the original sentence. At the same time, this paper verifies the robustness of the model by expanding the candidate paraphrase dictionary, proving that the model has good ductility. In general, the contributions of this paper mainly include the following four points: (1) the analysis of the advantages and disadvantages of existing methods is adopted to show the impact of the absence of a context-aware approach. (2) A context-aware approach is proposed to achieve better disambiguation by combining candidate paraphrases with acronym contexts. To the best of our knowledge, this is the first work to perform disambiguation from the sentence level. (3) The overall approach can simultaneously obtain sentence- and phrase-level similarity, which can get more information. (4) Experiments show that the proposed method can outperform the state-of-the-art methods on the public dataset when using the same BERT model.

2. Related Work

This section describes the acronym disambiguation methods based on dictionary and Siamese neural network.

2.1. Dictionary-Based Acronym Disambiguation

2.1. Dictionary-Based Acronym Disambiguation. Existing dictionary-based acronym disambiguation methods can be divided into five categories: feature matching (including statistics-based and classic machine learning-based methods), multiclassification, span prediction, binary classification, and similarity ranking [22].

2.1.1. Feature Matching Methods. The feature matching approach involves extracting features (e.g., discourse tags and special characters) from the input sentences. Statistical models are then used to predict the exact acronym interpretation. Statistics-based methods refer to the implementation according to the calculation formula of statistical word frequency and similarity, such as BM25 and TF-IDF. However, these methods usually cannot understand the semantic correlation between sentences and separate the semantics of words and sentences. It is inconsistent with the facts. With the development of machine learning, traditional machine learning methods based on maximum entropy, decision tree, and support vector machine are gradually emerging [14]. These algorithms are based on dictionary acronyms to eliminate discrimination as a classification problem. Maximum entropy aims to select the model with the most significant entropy among all possible probability models (probability distribution) [23]. However, the maximum entropy model will binarize the features, only record whether the features appear but cannot obtain the feature strength. The decision tree model is an attribute structure describing instance classification, mainly composed of nodes, and directed edges. The decision tree model usually starts from the root node, obtains the instance characteristics, and then assigns the instance to its child nodes [24]. The characteristics obtained by the decision tree model are easy to be affected by the amount of data. The support vector machine is to find the support vector that can determine the optimal classification hyperplane from the training samples by maximizing the classification margin [25]. The kernel function directly determines the performance of the support vector machine, but there is no suitable method to solve the problem of kernel function selection.

2.1.2. Multiclassification Methods. The multicategory problem is trained with each candidate interpretation as a category label. With the development of word vector models and neural network models, textual information has been able to be transformed into low-dimensional dense vectors. Current methods for acronym disambiguation are usually analysed on the basis of text embedding. This enables more contextual information to be obtained. The benchmark model GAD given by Veyseh et al. [9] is to obtain sentence embedding through Bi-LSTM and obtain context embedding with the help of grammatical structure (such as dependency tree) and GCN (graph progressive neural networks) model. Finally, the acronyms and sentence embedding under the two codes are spliced as the input of the evaluation layer, and then the interpretation of acronyms is predicted through a two-layer feedforward classifier. The number of neurons in the last classifier is equal to the number of candidate definitions of the acronym in the dictionary, but this also means that when the number of acronym definitions in the dictionary increases, the model structure will change significantly.

Jaber et al. combined three supervised machine learning models (support vector machine, naive Bayes, and k-nearest neighbor) with cosine similarity for acronym disambiguation among the feature-based methods. Finally, they found that the naive Bayes and cosine similarity method has the best performance [9]. Pereira et al. combined a support vector machine with the doc2vec method for acronym disambiguation [10]. These methods mainly extract the corresponding features from the text and predict the acronyms and corresponding interpretations by statistical methods. The neural network model challengers use mainly LSTM and CNN [26].

2.1.3. Span Prediction Methods. The transformer-based model mainly encodes sentences for BERT and its variants (such as Sci-BERT [15] or RoBERTa [27]). Still, there are differences in using the output of these language models to predict. Pan et al. [16] and Zhong [28] regarded the task as a classification task, while Egan and Bohannon [29] adopted the information retrieval method to calculate and sort the score of each candidate word by using the cosine similarity between candidate embedding and input. Singh and Kumar modelled the problem as a span prediction problem. It obtains the accurate interpretation from the connected text of acronyms, candidate interpretation, and sentence combination by the predicted probability of subsequence [14].

2.1.4. Binary Classification Methods. Binary classification is to combine the interpretation of a single acronym with the original sentence through the characteristic that BERT can process two sentences simultaneously [11]. The input format of two sentences is processed by simulating BERT, and the [CLS] identifier, candidate interpretation (according to the number in the dictionary), and [SEP] identifier are spliced with the original sentence as the model input and then train a binary classification model to acquire the score. This method is more robust and can handle longer dictionary lengths. However, this method does not consider the matching degree and correlation between the candidate interpretation and the original context. When doing acronym disambiguation, we can find the interpretation from the meaning of the acronym in the context and judge whether the interpretation conforms to the original context information.

2.1.5. Similarity Ranking Methods. The similarity ranking method specifically refers to the way of ranking by comparing the similarity scores of two inputs. Egan and Bohannon evaluated the similarity of the candidate paraphrases by directly comparing them with the original sentences and used the candidate with the highest similarity score as the predicted result [29]. This approach has similarity to the dichotomous approach. Nevertheless, the candidate interpretations contain limited information and the model may fail to evaluate when the two candidates themselves have similarity. In fact, there is complete substitutability between exact paraphrases and acronyms in contextual scenarios. In other words, replacing an acronym in the original sentence with an exact paraphrase will not change the meaning of the sentence at all.

Therefore, we propose a method to fuse the similarity of sentences with the similarity of the candidate translation itself. This approach can combine the candidate sentences with the context and can convey more features for the model. However, because of the limited text information that the model can handle, the length of the input text may exceed the upper limit that the model can handle better if the binary classification approach is used. We propose an acronym disambiguation method based on similarity ranking methods.

2.2. Siamese Neural Networks. Siamese neural networks, also known as Siamese networks, were first proposed by Bromley et al. [30] to verify the signature on the credit card. Now, it has been applied to many different fields, such as one-short learning [31], text recognition [29, 30], and face similarity recognition[32]. Unlike the traditional neural network model, the Siamese neural network model comprises two networks sharing weights. By transforming the two inputs into high-dimensional vectors and interacting with their features, the Siamese neural network model can realize the method of classification or similarity prediction. The advantage of a Siamese neural network is to identify the differences and similarities between the two inputs. That is, the Siamese network can measure the direct correlation degree of two inputs, in which network-1 and network-2 can be two same network models, such as CNN [33]or LSTM [34], transformer [35], or attention [36]. When the two networks do not share weights or utilize two different neural networks, such as an LSTM network and a CNN network, separately. We called this kind of models as called the pseudo-Siamese network [37]. With the development of BERT, Reimers N proposed to transform sentence pairs into two vectors with the same dimension through the same BERT model and then use different loss functions according to different tasks [38]. In the existing acronym disambiguation tasks, the methods' performance based on the pretraining language model is relatively higher than that based on features and traditional neural networks. Therefore, this paper will study the Siamese network based on BERT.

3. Limitations of Existing Methods

(1) Feature matching method: feature matching methods (including statistics-based and traditional machine learning-based methods) are prone to performance degradation and high cost in the face of large quantities of data. This type of method usually analyses only the number of occurrences of the acronym together with the paraphrase. If such methods tend to select cable news network as the CNN paraphrase, this selection is context-independent.

(2) Multiclassification method: the advantage of the multicategorization approach is the ability to select from multiple interpretations with only one calculation. However, the number of definitions of acronyms in the dictionary is often uncertain, and the number of categories is closely related to the shape of the last layer of the classification model. Therefore, the multiclassification method is easily disturbed by the number of candidate definitions of acronyms. For example, acronym "CA" has 20 paraphrases, which means the output dimension of the model is 20×1, while "RF" has only 5 paraphrases and the output dimension of the model is 5×1. This variability increases the difficulty of model training.

(3) Span prediction method: this method also attempts to perform interpretation recognition through a single computation. However, the input to the span prediction method is the acronym, all candidate paraphrases, and the concatenation of the original sentence, i.e., *[CLS] Acronym [SEP] Expansion_1 ... Expansion_N [SEP] Sentence [SEP]*, which means that the length of the input text is related to the candidate. The number of interpretations is directly related. Neural network models often need to perform zero-padded alignment processing on the input. When the length difference of each input is too significant, it is easy to cause the input mean and variance between different batches to be too different, which is not conducive to the robust processing of the model. Similarly, acronym "CA" has 20 paraphrases, which means that the input is the original sentence plus 20 paraphrases i.e., 40 words, while "RF" has only 5, adding only 10 words, which will introduce too much information when padding is used to compensate.

(4) Binary classification and similarity ranking algorithm: binary classification algorithms are the methods that splice candidate paraphrases with the original sentences, i.e., *[CLS] (Acronym [SEP]) Expansion_i [SEP] Sentence [SEP]* and then use a binary classifier to determine whether the paraphrase is correct or not. Also, similarity sorting is to sort the vector similarity between the original sentence and each candidate paraphrase, which is closer to the human way of thinking. Both methods are more robust and are not disturbed by the length of the lexicon. However, both ways do not consider whether the candidate paraphrases' match the context of the acronyms. The acronym candidate paraphrase should not only be semantically similar to the original acronym but also should be able to replace the acronym in the original sentence directly.

In the example in Table 1, the phrases that are semantically similar to the original sentence are random forest, regression function, and regression forest. But most of the models choose regression function or regression forest. While when asking the opinion of humans not working on machine learning, they mostly choose random forest. Because there is a tree in the original sentences, and they think that they should choose between random forest and regression forest. Also, since the RF is followed by a regression, it would be rather unusual for two identical words to appear next to each other in a sentence. They tend to choose random forest. According to this, we propose a context-aware method which analyses candidate paraphrases at both sentence and phrase levels.

4. Proposed Method

4.1. Problem Description. Given a sentence $S = [w_1, w_2, \ldots, w_n]$, the acronym position code is P, w_P is the acronym, the correct interpretation is d_i, and the acronym candidate interpretation dictionary is $D = \{w_P: d_1, d_2, \ldots, d_i, \ldots, d_s\}$, $d_i \in D$, where s is the number of definitions of each acronym in the dictionary. The acronym disambiguation task is to select the accurate interpretation d_i from the interpretation dictionary D according to the acronym w_P and sentence S. That is, the prediction of the model is $\operatorname{argmax}(p(d_i|S, w_P, D))$.

4.2. Overview of the Model. Different from the traditional methods of candidate interpretation for similarity evaluation with the original text, this paper will integrate the matching degree between the candidate acronyms and the acronym context at the same time. The acronym candidate replaces the acronyms in the original text to form a new sentence set and then takes the new sentence set and the original text as input to the Siamese network. The training is carried out by minimizing the loss function. If the label is 1 (similar), the embeddings of the two sentences are as close as possible. Otherwise, the distance between the two is as far as possible. This paper will construct sentence pairs from two levels: phrase and sentence levels. The first is to directly match the

Table 1: A confused example.

Raw sentence	Suppose θ is the parameter that determines a specific splitting node of RF regression trees
Raw dictionary	RF: ① random forest, ② radio frequency, ③ regression function, ④ regression forest, ⑤ register file

candidate interpretation with the sentence pair constructed by the original text, a single interpretation, as in [29]. The second is to match the new sentence formed by replacing the acronyms in the sentence with the candidate interpretation with the sentence pair composed of the original sentence, that is, the sentence pair. The specific examples are shown in Table 2.

It can be seen from Table 2 that in this experiment, there is no need to add any special characters to improve the attention of the model but only need to carry in two inputs directly into the model. The interpretation combination simulates the scene where the candidate's interpretation is directly compared with the original text. Through the direct encoding of the candidate interpretation, the encoding is transformed into a vector consistent with the encoding dimension of the original text through the pooling layer and then evaluated. Sentence combination is the result of comparing the new sentence formed by replacing the corresponding acronyms in the original text with the candidate interpretation. It is essentially the judgment between multiple sentences with the same context. The model needs to learn the differences from sentence perspective. The acronym disambiguation task constructed in this section belongs to the category of Siamese neural networks. We will evaluate the semantic similarity of two different sentence pairs by cosine similarity and take it as the score of the corresponding interpretation in the combination. The operation process is as follows: inputting s (the number of candidate definitions of target acronyms in the dictionary) sentence combinations and interpretation combinations into the Siamese network, respectively, and using cross-entropy loss as the loss function of interpretation combination and sentence combination, the code of each sentence pair is obtained.

Figure 2 shows the general framework of Siamese network structure based on BERT. We use interpretation combination to obtain the similarity score between the original sentence and the paraphrase and use sentence combination to obtain the similarity score between the original sentence and the paraphrase in the context-aware case. Finally, we use the weighted sum of the two as the final result. The SiameseNet in the figure represents the Siamese network model based on BERT, in which BERT mainly refers to the current commonly used BERT models, including BERT [12], RoBERTa [27], and Sci-BERT [15].

4.3. Siamese Neural Networks Based on BERT (SiameseNet).

We use the Siamese neural networks based on BERT [39] to evaluate the correlation between the acronym candidate and the original text, and the new sentence formed by replacing the acronym with the candidate and the original text. Then, we sort the interpretation of the candidate according to the two correlations. The candidate interpretation with the highest correlation is selected to be the answer. Siamese neural network structure can be divided into regression target structure and classification target structure according to different tasks, as shown in Figure 3.

In the figure, the model takes the two sentences (s_1 and s_2) as input into the BERT model for embedding and unify the sentence embedding dimension through the pooling layer to obtain two-sentence vectors u and v with the same dimension.

$$\begin{aligned} u &= \text{Pooling}(\text{BERT}(s_1)), \\ v &= \text{Pooling}(\text{BERT}(s_2)). \end{aligned} \tag{1}$$

Then, we augment the embedding by $|u - v|$, which means subtracting the two vectors u and v in element-wise and calculating the absolute value. The vectors u, v, and $|u - v|$ are concatenated into the fully connected layer, followed by the Soft Max layer, to obtain the final predicted score [40]. The objective function can be expressed as follows:

$$\begin{aligned} h &= \text{Concat}(u, v, |u - v|), \\ p &= \text{Softmax}(\text{FC}(h)), \end{aligned} \tag{2}$$

where FC means the fully connected layer.

We use the method based on classification and change the acronym disambiguation task into the classification task based on sentence similarity. In addition, different from the traditional methods of candidate interpretation for similarity evaluation with the original text, this paper will integrate the matching degree between the candidate acronyms and the acronym context at the same time. The acronym candidate replaces the acronyms in the original text to form a new sentence set and then takes the new sentence set and the original text as input to the Siamese network. The training is carried out by minimizing the loss function. If the label is 1 (similar), the embeddings of the two sentences are as close as possible. Otherwise, the distance between the two is as far as possible.

To enhance the robustness and generalization of the model, adversarial loss is used to train the SiameseNet network [39]. The objective loss function can be expressed as follows:

$$\begin{aligned} L_{\text{total}} &= \frac{1}{N}\sum_{n=1}^{N} \text{CE}(Y_n, p_n) + \beta \frac{1}{N}\sum_{n=1}^{N} L_{\text{adv}}^{n}, \\ L_{\text{adv}}^{n} &= Y_n \cdot p_n^2 + (1 - Y_n) \cdot \max(m - p_n, 0)^2, \end{aligned} \tag{3}$$

Table 2: The input form of model.

Raw sentence S	CNN is a kind of feedforward neural network
Raw dictionary D	CNN: ① convolutional neural network, ② condensed nearest neighbor, ③ complicated neural networks, and ④ citation nearest neighbor
Ground truth	Convolutional neural network
Interpretation combination	Interpretation combination 1 Input 1: CNN is a kind of feedforward neural network Input 2: convolutional neural network
Sentence combination	Sentence combination 1: Input 1: CNN is a kind of feedforward neural network Input 2: convolutional neural network is a kind of feedforward neural network

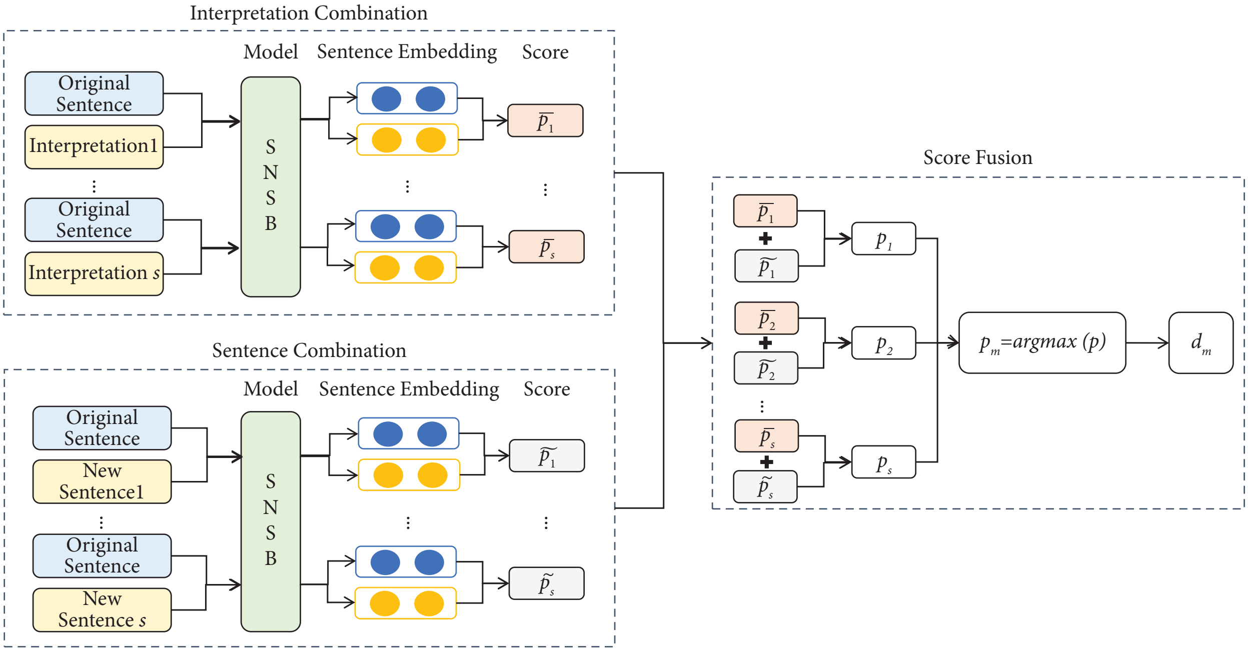


Figure 2: Illustration of the proposed ContextAD model.

where CE is the cross-entropy loss, which is used to portray the similarity between the actual output probability and the expected output probability. m represents the margin value, and N is the number of training samples. The value of Y is 1 or 0. If the two inputs are similar, it is 0, otherwise is 1. If the difference between the two inputs is less than the marginal value, the loss will be calculated; otherwise, the loss will be 0. In the process of adversarial training, this loss function can be split into a loss function when the samples are similar and a loss function when the samples are different, which can be expressed as follows:

$$L_{\text{adv}}^{n} = \begin{cases} p_n^2, & Y_n = 1, \\ \max\left(m - p_n, 0\right)^2, & Y_n = 0. \end{cases} \tag{4}$$

4.4. Score Fusion. We use the loss function of the formula (4) to train the SiameseNet included in the interpretation combination and sentence combining structures, respectively. After that, the two trained models are used for inferencing to obtain semantic similarity, and the score fusion is performed based on this, as shown in Figure 2. Assuming that the original sentence is s_{ori}, the interpretation statement of $s_{\text{int}} = [s_{\text{int}}^1, s_{\text{int}}^2, , s_{\text{int}}^s]$, the new sentence obtained by the combination is $s_{\text{new}} = [s_{\text{new}}^1, s_{\text{new}}^2, \ldots, s_{\text{new}}^s]$, the semantic similarity score of the word dimension is

$$\begin{aligned} \overline{p} &= [\overline{p}_1, \overline{p}_2, \ldots, \overline{p}_s], \\ \overline{p}_i &= \text{SiameseNet}\left(s_{\text{ori}}, s_{\text{int}}^i\right), \end{aligned} \tag{5}$$

where the output of SiameseNet is the SoftMax score in equation (2), which can be regarded as $\overline{p}(d_i | S, w_P, D)$, represents the similarity of the $\underline{i}$-th candidate paraphrase to the original sentence. Also, the score of the sentence dimension is

$$\begin{aligned} \tilde{p} &= [\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_s], \\ \tilde{p}_i &= \text{SiameseNet}\left(s_{\text{ori}}, s_{\text{new}}^i\right). \end{aligned} \tag{6}$$

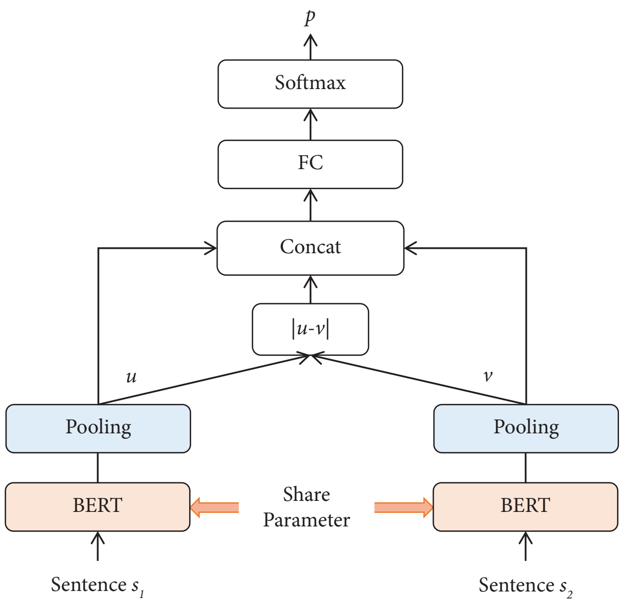


FIGURE 3: Siamese network structure based on BERT (SiameseNet).

Then, adding the scores of the word dimension and the sentence dimension, which is the final score

$$p = \alpha\overline{p} + (1-\alpha)\widetilde{p}, \tag{7}$$

where α is the weighting coefficient.

5. Experiments

5.1. Experimental Setup

5.1.1. Dataset and Benchmarks. We selected the AD data set in the SDU challenge in AAAI-2021 for experimental demonstration [9]. The original data include 50,034 training samples, 6,189 validation samples, and 6,218 test samples. However, because the label of the test sample is not disclosed and the challenging task has been closed, this paper will randomly select 10% of the data from 50,033 training samples as the verification set and the original verification set as the test set. Therefore, our experiment's number of training samples, verification set samples, and test samples are 45,031, 5,003, and 6,189.

The benchmark model framework of this paper includes the model of the dataset and single interpretation scoring, which are compared with sentence matching and double scoring. From the perspective of the model, this paper will select BERT-base, RoBERTa-base, and Sci-BERT-base, respectively, for experiments to provide a reference for follow-up research.

BERT refers to the BERT model initially proposed by Google. BERT-base uses 12 stacked embedding layers, each embedding layer uses 12 head attention, the feedforward network in embedding contains 768 hidden units, and the total parameters of the model are about 110 million [12].

The full name of RoBERTa is a robustly optimized BERT pretraining approach, which is a version of refined tuning of BERT [27]. The model mainly makes the following improvements to BERT: ① the dynamic mask method is adopted for model training, and the static mask is adopted for BERT, that is, the data are masked in advance, while RoBERTa adopts different mask modes when inputting sequences to the model, which means that the same data may have different mask modes in different epochs. RoBERTa believes that this method can teach more language representations. ② More training data, larger model parameters, larger batch size, and longer training time are used. ③ In RoBERTa, the next sentence prediction task is cancelled, and multiple sentences are input continuously until the maximum length is reached (cross text or not can be set, which is better in general). This means that the model can read longer text sequences simultaneously. This training method is called full sentences. ④ BERT uses Unicode characters as the subwords unit, with a size of about 30 K, while RoBERTa's embedding method combines character level and word level representation (BPE). This method includes 50 K subwords units without any additional preprocessing or word segmentation for input.

Sci-BERT is pretrained with a total of 1.14 million scientific papers in 82% biomedicine, 12% computer science, and 6% other disciplines [15], so it is more suitable for natural language processing tasks in the direction of scientific papers. In the SDU task of AAAI-2021, the data are collected from scientific and technological papers. Therefore, in the existing models, the effect based on Sci-BERT is often higher than that of other models. In addition, it can also be replaced with other network models, but the effect may be relatively poor. In this way, the relationship between the acronym and the interpretation, and the relationship between the context of the acronym and the interpretation of the acronym can be considered simultaneously, and the robustness can be taken into account. The model structure is independent of the length of the acronym dictionary and can deal with candidate dictionaries of any length.

5.1.2. Evaluation Protocol. In acronym disambiguation, precision, recall, and $F1$ score are usually used for evaluation.

$$\begin{aligned}\text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}},\\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}},\\ F1 &= \frac{2 * \text{precision} * \text{reccall}}{\text{precision} + \text{reccall}},\end{aligned} \tag{8}$$

where TP represents the number of entities predicted correctly, that is, the number of sequences whose predicted sequence is consistent with the real sequence, FP represents the number of entities predicted incorrectly, and FN represents the number of entities predicted incorrectly but actually correct.

5.1.3. Experimental Process. The experiment is divided into five steps: verification set division, dataset processing, model training, verification evaluation, and result evaluation. The specific operation contents are as follows.

Step 1: verification set partition. Since the submission channel of the original challenge task has been closed, it is necessary to extract 10% of the data from the training set as the verification set and the official verification set as the test set.

Step 2: data processing. Replace the corresponding acronyms in the original text with the candidate interpretation of acronyms to form a new sentence set and pair it with the original text. Among them, the sentence pair containing the accurate interpretation is labelled as 1, and the sentence pair where the other candidate interpretation is located is marked as 0 (a single interpretation is to pair the candidate interpretation directly with the original text to form a sentence pair, and the marking method is consistent with the sentence matching). In this way, it can also realize the expansion of the dataset in essence. The training set of the original dataset is 45,031, but the number of training set samples obtained by sentence matching is 203,438, which is expanded by 4.52 times. The number of samples of the original verification set is 5,003, while the number of samples of sentence matching is 22,779, which is expanded by 4.55 times. Also, the number of samples in the original verification set is 6,189, while the number of training samples is 28,286, which is expanded by 4.57 times. This processing method can have the effect of data enhancement.

Step 3: model training. According to different BERT models (mainly including three models: BERT-base, RoBERTa-base, and Sci-BERT-base), the training is carried out with the help of the sense transformer framework, and the loss function is a comparative loss.

Step 4: validation evaluation. The trained model is used to encode each sentence pair in the test set, and the cosine similarity is calculated. The candidate interpretation corresponding to the sentence pair with the highest cosine similarity is considered to be the correct interpretation. The values of α are 0, 0.1, 0.2, 0.3, ..., 0.9, 1.0, respectively.

Step 5: result evaluation. Results were evaluated using precision and recall and the harmonic mean $F1$ value of both, i.e., P, R, and $F1$ in the table. Also, the official ranking is mainly based on the macro $F1$ value. Because the current challenge task submission channel has been closed, this paper will directly compare with the Binary classification model on the verification set (the test set in this paper).

5.2. Experimental Results. The pretraining model in this paper mainly adopts huggingface (https://huggingface.co/models) and uses the sense transformer framework to build the model [22]. At the same time, this paper does not add any characters to the text or carry out any preprocessing to test the robustness of the model to unprocessed data. The experimental results are shown in Table 3.

Binary classification model in the table indicates the results obtained by the current state-of-the-art model on the official validation set (i.e., the test set of this experiment) using the corresponding pretrained model [16]. All other papers only have results from the test dataset, but we cannot get the results of our method on the original test set because the official access to it has been closed. It can be seen from Table 3 that under the same training conditions, the effect of sentence matching is significantly higher than that of using interpretation to match directly with the original document. The final scoring is the weighted sum of sentence matching and single interpretation. Experiments were conducted with values of α from 0.0 to 1.0, and the results of validation dataset showed that a sentence combination weight of 0.9 and a paraphrase combination of 0.1 worked best. Through the comparison of standard models, the dual scoring macro $F1$ value of Siamese network using Sci-BERT, that is, the $F1$ value of official ranking is the highest, reaching 91.965, 2.95% higher than that of the official embedding method. However, the relative effect of RoBERTa is poor. The reason may be that the model based on the Siamese network mainly obtains the embedding of sentence vector through fine-tuning, while RoBERTa's dynamic embedding mechanism and whole sentence training mechanism may lead to different concerns of the same sentence in different epochs, but the method of using Siamese network in the four models is better than that of the binary classification model.

5.3. Effect of the Fusion Hyperparameter. In the score fusion, there is an adjustable parameter α, which has a range of 0.0–1.0. It plays the role of weigh of sentence combination similarity. For the influence of the results, the experimental results are shown in the table. The classification results can be affected by a suitable adjustment parameter α. Figure 4 shows the results of $F1$ when α is set to various values. It has been found that when α is 0.9, the most accurate classification is provided. When α equal to 0, it represents the result of interpretation combination, when α equal to 1, it represents the result of sentence combination.

6. Further Analysis

6.1. Data Preprocessing. The statistical analysis of the equipped acronym dictionaries shows that the dictionaries contain a total number of 732 acronyms. The average number of interpretations of each acronym is about 3; the highest number of interpretations of each acronym is 20, and the lowest number is 2. Where 660 acronyms have less than five interpretations, accounting for 90.16% of the total number; 55 acronyms have between 5 and 10 interpretations, accounting for 7.51% of the total; 13 acronyms have between 10 and 15 interpretations, accounting for 1.78% of the total; and only four acronyms have more than 20 interpretations, accounting for 0.55% of the total, while the number of interpretations above 20 is only four acronyms, accounting for 0.55% of the total. The analysis of the test set revealed the number of samples containing these four acronyms, namely, "CA," "CS," "CC," and "SC." The number

Table 3: Experimental results.

Models		Precision	Recall	*F*1
Baseline (maximum word frequency) [22]		89.00	46.36	60.97
UC3M [10]		92.15	77.97	84.37
Acronym expander [26]		93.57	83.77	88.40
Human performance [22]		97.82	94.45	96.10
BERT-base	Binary classification [16]	91.76	81.60	86.38
	Interpretation combination [29]	89.91	71.64	79.74
	Sentence combination (ours)	93.16	82.43	87.47
	Score fusion (ours)	**93.37**	**82.66**	**87.69**
RoBERTa- base	Binary classification [16]	90.08	76.87	82.95
	Interpretation combination [29]	88.06	54.19	67.09
	Sentence combination (ours)	91.80	**78.63**	**84.70**
	Score fusion (ours)	**91.99**	78.00	84.42
Sci-BERT-base	Binary classification [16]	92.63	85.69	89.02
	Interpretation combination [29]	92.66	81.28	86.60
	Sentence combination (ours)	94.92	88.87	91.80
	Score fusion (ours)	**94.96**	**89.15**	**91.97**

The bold values in the figure represents the best result obtained by the corresponding BERT model under the same conditions.

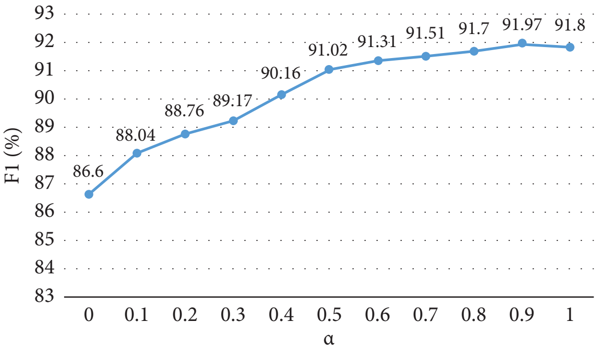


Figure 4: The *F*1 with different fusion hyperparameter.

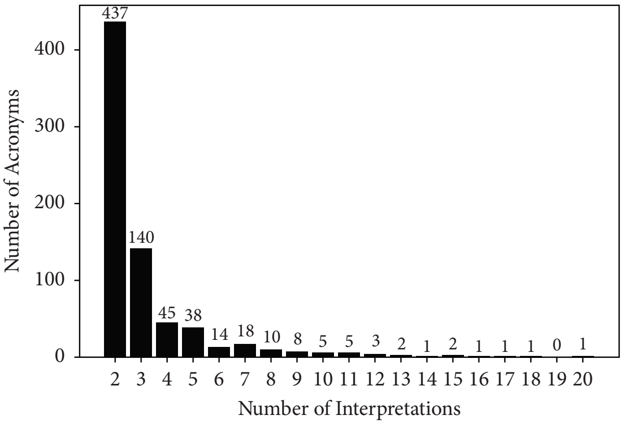


Figure 5: Overview of the number of interpretations of acronyms in the original dictionary.

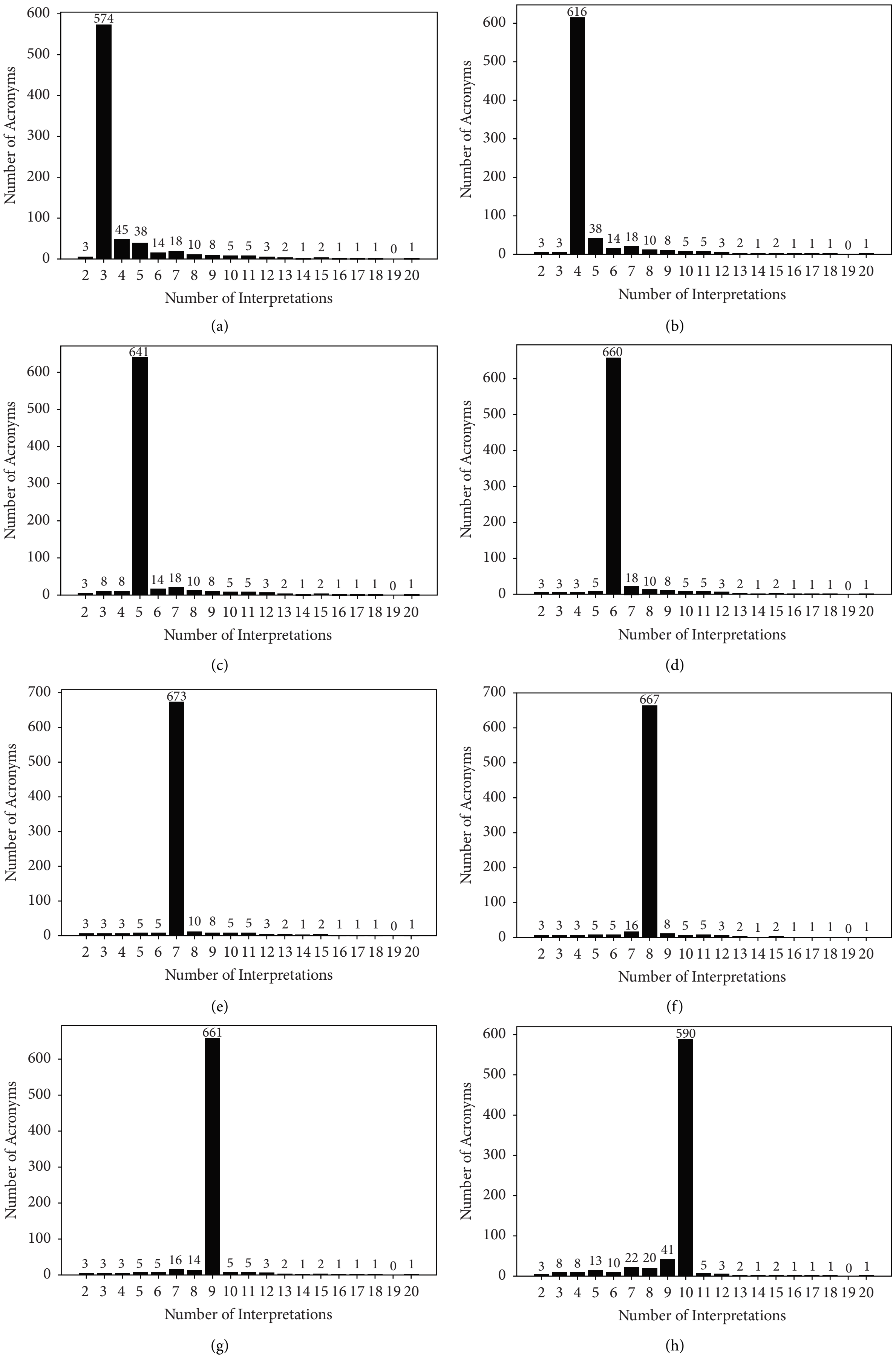
of samples containing these four acronyms, namely "CA," "CS," "CC," and "SC," was 44, 40, 35, and 18, respectively, accounting for only 2.21% of the total number of samples. Most of the samples contain the number of acronyms paraphrased concentrated in less than five. Moreover, these four acronyms should correspond to two-word phrases, and the acronyms for such phrases are not very meaningful but rather increase the difficulty of understanding the text. The specific overview is shown in Figure 5. However, the advantage of machines over humans is that they can process more information and data; so, this section will expand the existing lexicon based on the AcronymFinder (https://acronymfinder.com) website and conduct experiments to verify the robustness of the model.

From Figure 5, we can see that more than half of the acronyms in the lexicon are two quantities. Therefore, we will verify the sensitivity of the model by increasing the number of lexical acronyms with the help of the acronym website resource. The threshold of expansion is noted as *Num*, i.e., the number of acronym paraphrases less than *Num* is expanded to *Num*. The distribution of the expanded dictionary is shown in Figure 6.

Therefore, this section will be analysed using an extended lexicon as shown in Figure 6. First, only the test set and validation set are expanded. Also, the predictions are made using the model trained on the initial training set to demonstrate the sensitivity of the model. In the end, the entire dataset is expanded. Also, the model is retrained on the expanded training set for evaluation to demonstrate the expandability of the model.

6.2. Sensitivity Experiment Results. An overview of the dataset augmented according to the expanded dictionary is shown in Table 4.

From Table 4, when $Num = 3$, the test dataset will grow by 7.70%, i.e., the number of negative samples in test dataset grows by 7.70%. Also, when $Num = 6$, the test set will grow by a total of 50.23% of negative samples. When $Num = 10$, there will be an increase of 120.39%, and the ratio of positive to negative samples in the dataset will be nearly 1 : 9.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 6: The distribution of the expanded dictionary. (a) *Num* = 3. (b) *Num* = 4. (c) *Num* = 5. (d) *Num* = 6. (e) *Num* = 7. (f) *Num* = 8. (g) *Num* = 9. (h) *Num* = 10.

Table 4: Overview of the expanded validation and test dataset.

	Train	Validation	Test	Total	Growth rate of test dataset
Original dataset	45,031	5,003	6,189	11,192	—
Raw experimental data	203,438	22,779	28,286	254,503	—
Num = 3	203,438	24,590	30,463	258,491	7.70%
Num = 4	203,438	27,288	33,777	264,503	19.41%
Num = 5	203,438	30,602	37,886	271,926	33.94%
Num = 6	203,438	34,303	42,493	280,234	50.23%
Num = 7	203,438	38,111	47,231	288,780	66.98%
Num = 8	203,438	42,251	52,347	298,036	85.06%
Num = 9	203,438	46,582	57,671	307,691	103.89%
Num = 10	203,438	50,334	62,340	316,112	120.39%

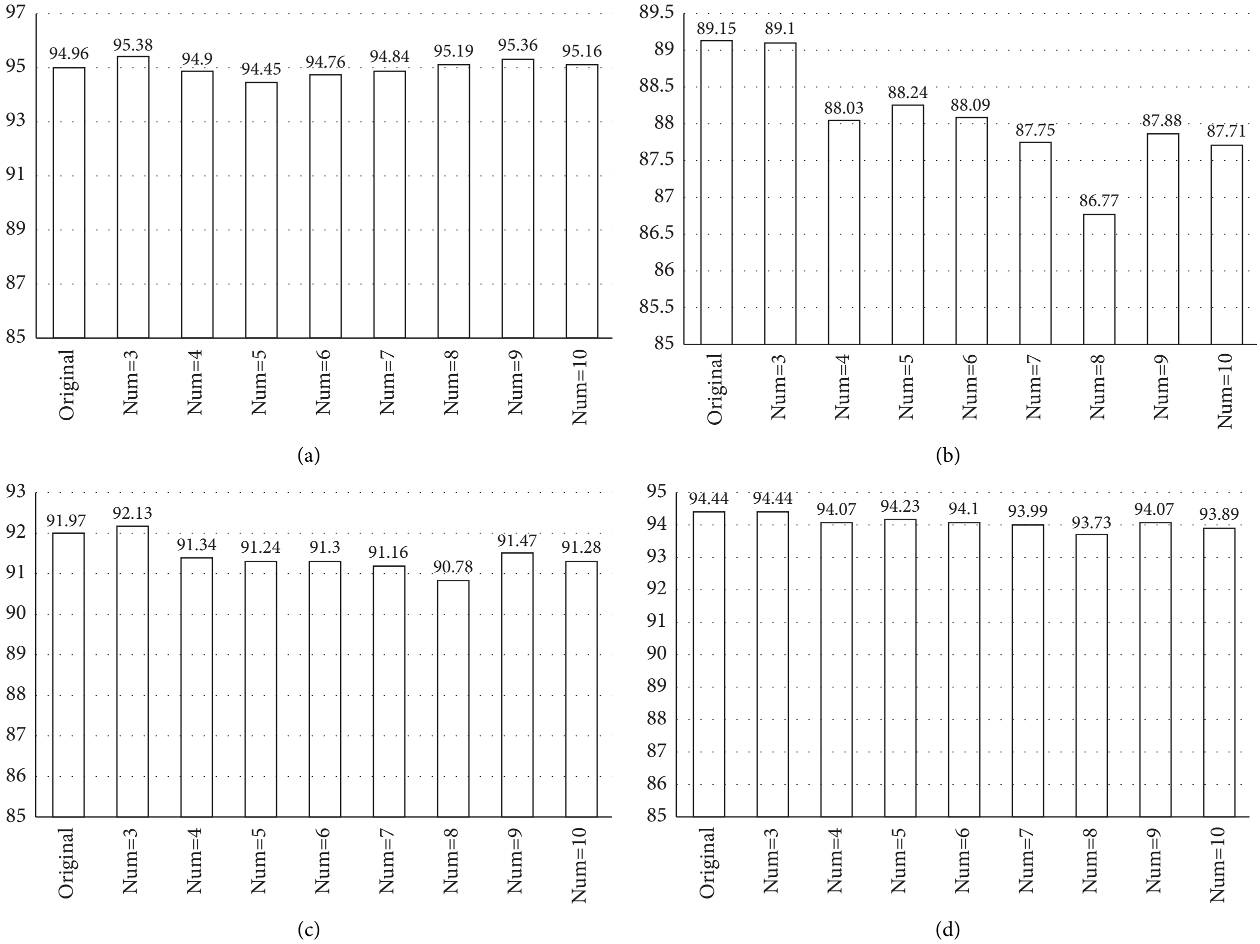


Figure 7: The result of the sensitivity analysis experiments. (a) Precision. (b) Recall. (c) $F1$. (d) Accuracy.

This experiment uses the Siamese network framework based on Sci-BERT with the highest $F1$ value for the experiment, the epoch of the model is 4, the batch size is 16, and the maximum length of the encoding is 400. Data validation is performed once every 500 batch sizes, and the model with the best performance in the validation set is retained. The specific experimental results data are shown in Figure 7.

Overall, the model effect fluctuates with *Num* changes. Although there is an overall decreasing trend, the overall amount of fluctuation is within 2%. In interpretation combination, the recall value is the lowest, followed by the $F1$ value, while precision is the best. It is noteworthy that both precision and $F1$ values achieve the maximum value at *Num* = 3 when the data growth rate of the test sets is 7.70%.

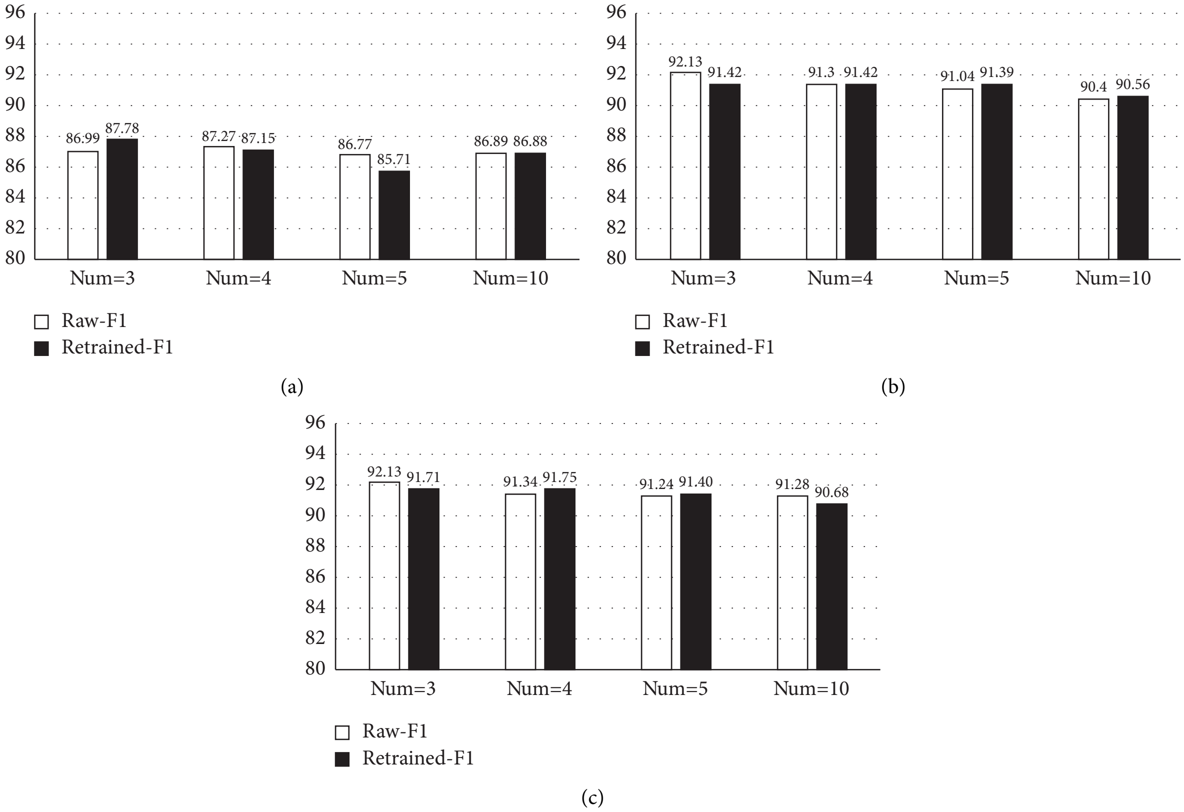
Table 5: Comparison of statistical indicators of $F1$ value.

Models	Average	Range	Variance
Interpretation combination	86.85	1.54	0.21
Sentence combination	91.00	1.79	0.41
Score fusion	**91.41**	**1.35**	**0.17**

The bold values in the figure represents the best result obtained by the corresponding BERT model under the same conditions.

Table 6: Overview of the expanded dataset.

	Train	Validation	Test	Total	Growth rate
Original dataset	45,031	5,003	6,189	56,223	—
Raw experimental data	203,438	22,779	28,286	254,503	—
$Num = 3$	220,136	24,590	30,463	275,189	7.70%
$Num = 4$	244,863	27,288	33,777	305,928	19.41%
$Num = 5$	275,234	30,602	37,886	343,722	33.94%
$Num = 10$	453,200	50, 334	62, 340	565, 874	50.23%



(a)

(b)

(c)

Figure 8: The $F1$ values comparisons of raw and retrained models. (a) Interpretation combination. (b) Sentences combination. (c) Score fusion.

The mean, range, and variance distribution of the $F1$ values of the three models are shown in Table 5.

Most existing researches directly compare the candidate's paraphrases with the original sentences. That is, the interpretation combination is used. However, it can be seen from Table 5 that the interpretation combination has the disadvantage of a lower $F1$ score than the sentence combination, but it is more stable. Both the range and variance are lower. The score fusion combines the two advantages: a higher average $F1$ value, lower range and variance, and more stability. Therefore, using the score fusion is more robust and efficient than existing interpretation combination methods.

6.3. Scalability Experiment Results. Scalability experiments are performed on an expanded training set using the same model with the same parameters and environment. An overview of the dataset after expanding the training set is shown in Table 6.

It can be seen from the table that after the training set is expanded, the growth rate of the total sample size is similar to that of the test set. However, with the same batch size = 16,12,715 iterations are required in a single epoch in the original training data, and when training with the RTX 3090, the duration of a single epoch is about 43 minutes (only the duration of the first epoch is recorded). When using the expanded dataset, when *Num* = 3, a single epoch requires 13759 iterations, and the training time of a single epoch is about 57 minutes 500 iterations are set for one validation during training, with the number of validation sets increasing. When *Num* = 4, a single epoch takes about 1 hour and 27 minutes; when *Num* = 10, a single epoch takes about 5 hours and 2 minutes, and four epochs will take about 5 hours and 2 minutes. For more than 20 hours, the consumption of electricity and computing resources is enormous. The *F*1 values result comparison of original model and the retrained model on the expanded dataset is shown in Figure 8.

The time and resource cost of retraining is several times the training cost of the original model, but as can be seen from the Figure 8, the retrained model results are very close to the actual model results or even worse than the initial model results. This means that the model has good scalability. In practical, a small-scale dictionary can be used for training and then applied to a large-scale dictionary to save resources.

7. Conclusion

In this paper, we propose ContextAD, a context-aware similarity ranking method, which mainly exploits the feature of complete substitutability between exact paraphrases and acronyms. ContextAD mainly performs ranking prediction by comparing the similarity between new sentences containing candidate paraphrases and the original sentences containing acronyms. Then, a score fusion method is designed to weight and rank candidates according to the similarity score of the interpretation and sentence combination, to improve performance and robustness. The experiments results show that the model does not require additional trained models and data to achieve results beyond SOTA. In addition, we also design an experiment to extend the number of acronyms paraphrases, which effectively verifies the robustness of the model.

In future work, we will conduct further research from two aspects. (1) Multilingual applications, acronyms are not unique to English, but Chinese (Pinyin), Spanish, and French all have this phenomenon. Therefore, we will carry out multilingual or even cross-lingual disambiguation to better understand scientific literature. (2) Large-model generative disambiguation. With the development of large-scale generative language models, we will study disambiguation methods that directly generate acronyms paraphrases.

Data Availability

The datasets and evaluation scripts can be accessed through the following link: https://github.com/amirveyseh/AAAI-21-SDU-shared-task-2-AD. Also, the other supplementary data are described in the article. All of them are publicly available.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61906207).

References

[1] K. Jacobs, A. Itai, and S. Wintner, "Acronyms: identification, expansion and disambiguation," *Annals of Mathematics and Artificial Intelligence*, vol. 88, no. 5–6, pp. 517–532, 2020.

[2] E. Sumita and F. Sugaya, "Using the web to disambiguate acronyms," *Human Language Technology of the NAACL*, vol. 6, pp. 161–164, 2006.

[3] A. Jain, S. Cucerzan, and S. Azzam, "Acronym-expansion recognition and ranking on the Web," in *Proceedings of the 2007 IEEE International Conference on Information Reuse and Integration 2007*, pp. 209–214, Las Vegas, NV, USA, August 2007.

[4] J. Pustejovsky, J. Castaño, B. Cochran, M. Kotecki, and M. Morrell, "Automatic extraction of acronym-meaning Pairs from MEDLINE databases," *Studies in Health Technology and Informatics*, vol. 84, no. 2, pp. 371–375, 2001.

[5] M. Roche and V. Prince, "A web-mining approach to disambiguate biomedical acronym expansions," *Inform*, vol. 34, no. 2, pp. 243–253, 2010.

[6] M. Roche, "How to exploit paralinguistic features to identify acronyms in text," *ACL-ISO Work*, vol. 12, pp. 69–72, 2014.

[7] D. H. Jeong, J. Gim, and H. Jung, "Incremental discriminating method for acronyms in heterogeneous resources," *International Journal of Advances in Soft Computing and Its Applications*, vol. 7, no. 1, pp. 59–67, 2015.

[8] Y. Li, B. Zhao, A. Fuxman, and F. Tao, "Guess me if you can: acronym disambiguation for enterprises," *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 1308–1317, 2018.

[9] A. P. Ben Veyseh, F. Dernoncourt, T. H. Nguyen, W. Chang, and L. A. Celi, "Acronym identification and disambiguation shared tasks for scientific document understanding," *CEUR Workshop Proc*, vol. 2831, 2021.

[10] A. Jaber and P. Martínez, "Participation of UC3M in SDU@ AAAI-21: a hybrid approach to disambiguate scientific acronyms," *CEUR Workshop Proc*, vol. 2831, 2021.

[11] W. Rogers, A. Rae, and D. Demner-Fushman, "AI-NLM exploration of the Acronym identification shared task at SDU@AAAI-21," *CEUR Workshop Proc*, vol. 2831, 2021.

[12] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding," *Proceedings of naacL-HLT*, vol. 1, p. 2, 2019.

[13] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: improving pre-training by representing and predicting spans," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020.

[14] A. Singh and P. Kumar, "SciDr at SDU-2020: IDEAS Identifying and disambiguating everyday acronyms for scientific Domain," *CEUR Workshop Proc*, vol. 2831, 2013.

[15] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: pretrained contextualized embeddings for scientific text," 2019, http://arxiv.org/abs/1903.10676?utm_source=researcher_app&utm_medium=referral&utm_campaign=RESR_MRKT_Researcher_inbound.

[16] C. Pan, B. Song, S. Wang, and Z. Luo, "BERT-based acronym disambiguation with multiple training strategies," *CEUR Workshop Proc*, vol. 2831, 2021.

[17] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: decoding-enhanced bert with disentangled attention," 2020, https://arxiv.org/abs/2006.03654.

[18] P. He, J. Gao, and W. Chen, "Debertav3: improving deberta using electra-style pre-training with gradient-disentangled embedding sharing," 2021, https://arxiv.org/abs/2111.09543.

[19] Y. Weng, F. Xia, B. Li, X. Huang, and S. He, "Adbcmm acronym disambiguation by building counterfactuals and multilingual mixing," *CEUR Workshop Proceedings*, vol. 3164, 2022.

[20] C. Raffel, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, pp. 1–67, 2020.

[21] G. Song, H. Lee, and K. Shim, "T5 encoder based acronym disambiguation with weak supervision," *CEUR Workshop Proc*, vol. 3164, 2022.

[22] A. Pouran Ben Veyseh, F. Dernoncourt, Q. H. Tran, and T. H. Nguyen, "What does this acronym mean? Introducing a new dataset for acronym identification and disambiguation," in *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 3285–3301, Barcelona, Spain, December 2020.

[23] V. P. Singh, *Entropy and Principle of Maximum Entropy*, Springer, Berlin, Germany, 1998.

[24] B. Chen, Q. Chen, and P. Ye, "Information-based massive data retrieval method based on distributed decision tree algorithm," *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 14, no. 01, 2023.

[25] X. Huang, S. Zhang, C. Lin, and J. Xia, "Quantum fuzzy support vector machine for binary classification," *Computer Systems Science and Engineering*, vol. 45, no. 3, pp. 2783–2794, 2023.

[26] J. L. M. Pereira, H. Galhardas, and D. Shasha, "Acronym expander at SDU@AAAI-21: an acronym disambiguation module," *CEUR Workshop Proc*, vol. 2831, 2021.

[27] Y. Liu, *RoBERTa: A Robustly Optimized BERT Pretraining Approach*, Vol. 1, Springer, Berlin, Germany, 2019.

[28] Q. Zhong, "Leveraging domain agnostic and specific knowledge for acronym disambiguation," *CEUR Workshop Proc*, vol. 2831, 2021.

[29] N. Egan and J. Bohannon, "Primer AI's systems for acronym identification and disambiguation," *CEUR Workshop Proc*, vol. 2831, 2021.

[30] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *Advances in Neural Information Processing Systems*, vol. 6, 1993.

[31] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," *ICML deep learning workshop*, vol. 2, p. 1, 2015.

[32] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 539–546, Diego, CA, USA, June 2005.

[33] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," *Lecture Notes in Computer Science*, vol. 9914, pp. 850–865, 2016.

[34] S. Bhati, L. M. Velazquez, J. Villalba, and N. Dehak, "LSTM siamese network for Parkinson's disease detection from speech," in *Proceedings of the IEEE GlobalSIP 2019 7th IEEE Global Conference on Signal and Information Processing 2019 conference proceedings*, Shaw Centre, Ottawa, Canada, November 2019.

[35] H. Cheng, B. Rao, L. Liu et al., "PepFormer: end-to-end transformer-based siamese network to predict and enhance peptide detectability based on sequence only," *Analytical Chemistry*, vol. 93, no. 16, pp. 6481–6490, 2021.

[36] H. Zhang, D. Hu, and Y. Qiu, "A siamese network tracking algorithm based on hierarchical attention mechanism," *Journal of Physics: Conference Series*, vol. 1828, p. 12044, 2021.

[37] C. Wang, S. Ge, Z. Jiang, H. Hao, and Q. Gu, "SiamFuseNet: a pseudo-siamese network for detritus detection from polarized microscopic images of river sands," *Computational Geosciences*, vol. 156, Article ID 104912, 2021.

[38] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proceedings of the Similarity-Based Pattern Recognition: Third International Workshop SIMBAD 2015*, vol. 3, pp. 84–92, Copenhagen, Denmark, October 2015.

[39] N. Reimers and I. Gurevych, "Sentence-BERT: sentence embeddings using siamese BERT-networks," in *Proceedings of the EMNLP-IJCNLP 2019 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, pp. 3982–3992, Hong Kong, China, November 2019.

[40] M. Al Duhayyim, H. Mesfer Alshahrani, F. N Al-Wesabi, M. Alamgeer, A. Mustafa Hilal, and M. Ahmed Hamza, "Relation-aware entity matching using sentence-bert," *Computers, Materials and Continua*, vol. 71, no. 1, pp. 1581–1595, 2022.